A High-Density GaAs Gate Array Chip Set for a Deep Space Digital Receiver

Dr. Gary R. Burke
Radio Frequency and Microwave Subsystems Section
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109

NASA's Deep Space Antennas

The Jet Propulsion Laboratory (JPL), a division of the California Institute of Technology, performs research, development, and related activities for the National Aeronautics and Space Administration (NASA). JPL is NASA's lead center for unmanned exploration of the solar system and beyond, As part of this effort, JPL designs, engineers, and manages the Deep Space Network (DSN), NASA's worldwide antenna system for communicating with spacecraft traveling beyond Earth orbit. The DSN is the largest and most sensitive telecommunications and radio navigation network in the world. The three DSN sites are located in Goldstone, California, near Madrid, Spain, and near Canberra, Australia. The stations are separated approximately 120 degrees in longitude so that as the Earth turns on its axis, a distant spacecraft is always in communication with one of the net work's stations. The DSN has more than a dozen antennas, ranging in diameter from **26** meters **(85** feet) to **70** meters (230 feet), Each DSN antenna has its own transmitter/receiver. Figure 1 is a photograph of the Goldstone 70m antenna.

Communications in Deep Space

Contact is still maintained with Pioneer 10, launched over 30 years ago and now at the outer edges of our solar system. Pioneer 10 continues to return valuable data about the boundaries of the heliosphere (the edge of the Sun's gravitational and magnetic influence). The communication link with Pioneer 10 is S-band (2110 to **2300** MHz). The spacecraft signal is 8 watts and travels a distance of about 60 AU (nearly 5 billion miles). The telemetry data rate of Pioneer 10 is 16 bits/second.

Since the days of the Pioneer missions, two additional frequency bands have been added that permit a faster data rate and higher power transmission: X-band (8400 to 8450 MHz) and recently, Ka-band (32 GHz). There are four uses for the communications link to the spacecraft. First, the spacecraft signal is used to correctly point the antennas (tracking)--a ranging code from the spacecraft indicates its distance from the antenna, and the doppler shift of the signal indicates its velocity. Second, very high-power signals are transmitted to the spacecraft for navigation and control. Third, the quality of the spacecraft signal is analyzed for information about the media (e.g., a planetary atmosphere) through which the signal has passed (radio science). Fourth, the signal is used to transmit data from the spacecraft's sensors (telemetry); this information includes images from cameras as

well as instrument readings, and is transmitted in binary code.

Current telemetry rates for the various NASA and other agency spacecraft vary from 8 symbols per second to 3 megasymbols per second. Each symbol is (normally) 1 bit of information. The spacecraft signals which reach the DSN receivers are faint and hidden in background noise. To filter out the signals from the noise, the receivers contain advanced signal processors.

A New Deep Space Receiver

The Block V receiver will replace several different types of analog receivers. The existing receivers are 20-30 year old designs.

Future missions, such as the Mars mission, will require high data rates, over 20 Megasymbols per second. The DSN's current generation of analog receivers are limited to 4 Megasymbols per second. The Block V receiver will use digital signal processing techniques to increase its performance, and will be capable of meeting DSN tracking and telemetry requirements until 2010 and beyond. In addition to increased sensitivity, the receiver will be capable of telemetry bandwidths of 66 MHz (the current limit is 4 MHz) and improved minimum tracking loop bandwidth of 0.1 Hz (the current limit is 1.5Hz).

The improved minimum tracking loop bandwidth of the Block V receiver will allow it to track Pioneer 10. This spacecraft is already out of the range of the analog receivers. The receiver may also be required to play a vital role in increasing the data return rate of the Galileo spacecraft, now on route to Jupiter, The spacecraft developed problems with its telemetry antenna after its launch in October 1989. Galileo has two antennas, a small low-gain antenna, which is functioning normally, and a larger high-gain antenna, which functions like an umbrella and was stowed in a folded position at launch, The unfurling in space of Galileo's high-gain antenna did not go smoothly, and two of the ribs on one side are stuck in a semi-opened position. If the antenna cannot be fully released before it arrives at Jupiter, the spacecraft will be forced to rely on its low-gain antenna for all communications. This fallback procedure will necessitate a great deal of data compression, and will be dependent upon the Block V Receiver's ability to provide a reasonable data rate. Galileo will arrive at Jupiter in December 1995.

The Block V Receiver

Figure 2 is a simplified block diagram of the Block V Receiver. An 8-bit digital signal from the antenna is amplified and then translated in frequency to 200 MHz. This signal is next sampled by an A/D converter at 160 MHz and fed into the digital portion of the Block V Receiver. The signal consists of the carrier and the data modulation. The carrier is digitally demodulated to provide the subcarrier, and then further demodulated to provide the symbol information.

The large amount of digital signal processing that must be performed makes an ASIC considerably more attractive than a discrete implementation. However, the 160-MHz

rate would be difficult to achieve in CMOS. Also, the pressing design deadlines for the Block V Receiver precluded a custom solution, dictating instead the use of gate arrays.

ECL gate arrays could easily meet the 160-MHz clock rate, but were not available in very high integration levels and also consumed high amounts of power. The choice of GaAs was made for the ASICs because of the high density of the gate arrays (350k gates), the modest power levels (180 microwatt per gate), and the ability with GaAs to achieve a pipeline speed of 160 MHz.

Three ASICs are used: a carrier ASIC (Jetcar) performs the initial demodulation, a subcarrier ASIC (Jetsub) processes the subcarrier, and a symbol ASIC (Jetsym) processes the symbol information; since two channels may be carried on one carrier, the digital demodulator has two subcarrier ASICs and two symbol ASICs, plus one carrier ASIC, for a total of five ASICs.

Digital Demodulation with ASICs

Figure 3, a block diagram of the digital demodulator, shows the variety of tasks that the ASICs must perform. The sampled IF signal (8-bit data) is first fed into the carrier ASIC (Jetcar). The Jetcar contains a quadrature demodulator (carrier demodulator) that includes two 8x 16 multipliers. The carrier oscillator contains a numerically controlled oscillator (NCO). This produces a ramp function, which is converted to sine and cosine values by random logic. These values are the other inputs to the multipliers. For easier implementation on the gate array, logic gates were used for this function rather than ROM.

The multipliers are followed by two lowpass filters, The multiplier/add operation of these filters is implemented by a merged Wallace tree-type adder array. The frequency of the NCO is updated at a 10-kHz rate by software. The output of the filters is sampled at 80 MHz. This represents the inphase and quadrature components of the subcarrier and is fed into the subcarrier ASIC.

An accumulator performs a decimate and sum operation on the quadrature output, reducing it to 10 kHz. This is read by software, from which a phase error is calculated. The phase error is used to update the frequency of the NCO. This process is typical of many phase-locked loops in the system, which are closed by a combination of hardware and software, The Jetsub and Jetsym ASICs operate at an 80-MHz rate. The Jetcar ASIC is supplied with an accurate 160-MHz clock and generates the 80-MHz clock for the other ASICs.

The Jetsub ASIC includes logic to demodulate the subcarrier with quadrature demodulation and filtering. The Jetsym ASIC includes symbol loop processing with NRZ and Biphase decoding blocks. It contains accumulators set up to perform weighted integration over the symbol "window." Subcarrier and symbol loops are closed in software. A VME bus is used to connect the ASIC with the CPU running the software. The final

symbol data rate is low enough so that the data can be read over this bus. High-speed monitoring is also performed over an additional bus (not shown in Figure 3).

Designing the ASICs

It was estimated that about 150k gates were needed per ASIC. The Vitesse 350k gate array family was chosen because the Vitesse is a channelless array with typically 50% utilization, making it a good fit for this application. The gate array allows use of both ECL and TTL signal levels; this was useful since the VME interface was lull level but the signal data and clocks were ideally ECL levels. Fig 4 is a photograph of the Jetcar ASIC. (The photo shows the chip with only two layers of metal; the final device has four metal layers, but when these layers are added, the chip structure is no longer visible.)

To minimize design risk, a fixed clock tree was used. The clock tree is built into the master slice and guarantees that there are no load violations between registers. The drawback is that only one clock may be used for the array. In the case of the Jetcar ASIC, the 160-MHz sections are fed directly from the clock, while the 80-MHz sections are also connected to the 160-MHz clock but are enabled by a 80-MHz signal derived from the clock. This signal allows data to change only at the lower rate. Similarly in the other chips, some of the logic operates at two-clock or three-clock intervals, but is still connected to the global clock.

Designing the circuits to operate at 160 MHz is a problem, even with GaAs. Consider the NCO circuit output register (Figure 5). This circuit consists of a 40-bit adder, an output register, a frequency register, and a phase register. The registers are all 40 bits wide. During startup an initial phase is loaded into the adder; during normal operation the contents of the frequency register are added to the accumulator at every clock interval. An update cycle can change both the frequency and phase.

The 160-MHz clock rate leaves only about 5 nanoseconds for the worstcase path through the adder after allowing for clock skew, worstcase processing, and safety margin. (An additional safety margin of 15% was used on all timing.) Full carry lookahead is used in the adder to achieve this speed. The adder is constructed out of 4-bit macros, which are used to create high-speed 16-bit and 8-bit adders. The 40 bits actually comprises two 16-bit adders and an 8-bit adder, with full carry lookahead. The registers are constructed out of 8-bit registers, each with its own driver for scan enable. Each register block has full scan. The scan-out is buffered so that it does not load the 1‘s bit of the register.

The multipliers in the carrier demodulator also run at 160 MHz. These multipliers are designed with Wallace tree adders and full carry lookahead final adders. The only pipeline stage is inserted between the Wallace tree array and the full adder stage. There are many different types of multipliers in the three ASICs. To simplify the design process, the multipliers were built out of reusable building blocks. These same building blocks were used for the squaring circuits and the digital filters.

Strategies and Tools for Chip Layout

The most challenging part of this ASIC design task was ensuring that the chip set would work at the specified speed. The problem is that the speed of the gates is not known until after layout. The gate speed is highly dependent on the wire load on the gate. The GaAs NOR gate is the basis for all the GaAs logic. Figure 6 shows the structure of the NOR gate. In the basic gate, the output pull-up transistor is passive. This results in a slow rise time, but variations to this gate buffer the output to increase the output drive. In the Vitesse logic family, three drive strengths are available: 0.5x, 1x and 2x, Figure 7 shows the effect of loading on three types of GaAs NOR gates with 0.5x, 1x, and 2x drives.

Because of the interdependence of gate speed with wire length, the best layout for high performance is achieved by hand-placing all the macro-cells (replacement). In this case preplacement was impractical since it would significantly impact the project schedule. The design of each block is dependent on the layout--for example, additional pipeline stages or additional buffering can be added to improve speed if necessary. Front annotation techniques will estimate the wire length before placement, but not very accurately. A technique called Intelligent Front Annotation was used to give a realistic estimate of wire lentgh. Figure 8 shows the design strategy used, By using Intelligent Front Annotation (described below) in conjunction with a Static Timing Analyzer, each block was designed to run at the design speed (plus margin). These design tools ensured that the final block would perform at the correct speed.

Modern placement tools can do a reasonable job of placement, given the correct constraints. The estimates from Intelligent Front Annotation can also be used as constraints for the placement program. The technique is illustrated in Figure 9. EXAM, a statistical analysis tool, is used to qualify the placement run. If it passes, the static analyzer is run using the Back Annotation values to check all delay paths, Additionally, the design is run at final speed using the logic simulator.

Intelligent Front Annotation (IFA) was also used to generate a good estimate of the wire lengths before placement. This tool (which was written at JPL for this project) uses the design hierarchy to estimate the distance between the macro-cells. For example, a small hierarchical block is assumed to be placed in the same area of the chip so that the wire lengths can be estimated based on the size of that block. Interblock networks are assumed to lie in the hierarchical level above those blocks, and are allotted their wire length based on that level. This process becomes complicated when nets have multiple fanouts and travel between many hierarchical levels. In this case the IFA program determines the most likely order of the nodes in the net, and then calculates the sum of the estimated distances between the nodes.

Placement efficiency is aided by using floor planning, and by the use of soft-groups. Floor planning assists the placement program by dividing the task into smaller, more manageable units. Each chip is divided into regions. Each macro-cell is associated with one region only, and can only be placed within that region. However, region

boundaries can overlap. Within each region, the design is divided into soft-groups. Each soft-group contains a block of logic which belongs together (e.g., an adder or multiplier). The region and soft-group information was captured in the design by attaching properties to blocks on the schematics. The IFA tool also uses the region and hierarchical information to generate a constraints file for the Place and Route programs.

For debugging and to prove that good placement could be achieved, two test cases were designed. These two designs represented one region of the Jetcar chip. These test cases were generated in a very short time, and many iterations of place and route were performed on them by JPL with the cooperation of Vitesse. The EXAM program was used to evaluate the placement and the results were used to refine the IFA program. By the time the first real chip was run through placement, the method was refined to the point where only two placement runs were required per chip.

Design Tools

Like any other ASIC technology, standard design tools were used for the gate array design. A static timing analyzer, ATPG software, and simulation accelerator were added to the CAE software to help with the design and verification. All these tools were directly (or indirectly) supported by libraries from Vitesse, together with translators and ERC checking tools. Static timing analysis and ATPG software were used on the individual blocks of the circuit to verify timing and testability before assembling them into the chip; this saved time in the final chip simulation process.

It was realized early in the project that a software simulation of the entire chip would be very slow. Therefore, a simulation accelerator was used to speed up simulation time by a factor of over 100. It was also realized that it would be impossible to predict the worstcase path between pipeline stages in many cases. The static analyzer circumvents this problem by verifying every path between pipeline registers.

Table 1 summarizes the tools used for the design. Figure 10 shows a block diagram of the design process, Functional verification is achieved in two ways: First, a C model is constructed for each block and the simulation outputs are compared with the expected values from the C model, This is especially useful to verify outputs of arithmetic blocks such as multipliers, which are easily simulated by the C model. Next, at the top level, the chip outputs are compared with the outputs of a signal processing simulator which models the chip at a much higher functional level, and which incorporates all of the signal loops, including the software components.

Testability

Testability y was a concern for several reasons. The chips were large and constructed using new technology. A fault on a chip may not show up easily in the system due to inherent fault tolerance: The receiver is designed to extract a signal buried in noise; a chip defect could in effect just add to this noise, and in that case the receiver would still function, but not as well as it should, It is clear that a set of test vectors must be

produced that is capable of detecting any manufacturing defect, In practice, this is not possible, and an attainable goal was set,

The test vectors are constructed to detect a high percentage of "detectable single stuck at faults." This was achieved by using the fault grading capability of the simulation accelerator, and by using the ATPG tool.

Each block is individually tested for testability (see Figure 11). The set of functional vectors used to prove the simulation model is also employed as an initial set of structural vectors, To determine the fault coverage (the percentage of faults detected by test vectors), a fault simulation is run. If the coverage is insufficient, additional vectors can be produced by hand, or generated automatically using the ATPG program. To save time, an in-house translator (FLP) was used to port the fault list from the accelerator to ATPF and vice-versa. This was considerably faster than re-running the functional test vectors in the ATPG environment. Additionally, the ATP fault list is used to generate a reduced fault list for the Fault simulator by re-running FLP. This reduces the final fault simulation time. The fault-list is toggled back and forth in this way until high coverage is achieved,

To assist in generating high-coverage test vectors, internal registers were connected to a scan path. Although ATPG can work in partial scan designs, it was found to be too slow. With full scan, the ATPG would achieve over 99% coverage in a few hours. In the ASIC designs, the number of scannable flip-flops was very large (2000-2400 bits). Rather than create one huge scan path, these were divided into multiple scan paths of 128 bits each. This does not affect fault coverage, but does considerably reduce the number of test vectors required.

Boundary Scan was included in the ASICs. This function was separate from the internal scan, and was not used for chip testing, Mainly, it is required for board testing, where patterns are transmitted from chip to chip to verify board interconnections. The JTAG-compatible boundary scan was used in order to take advantage of JTAG macro-cells in the Vitesse library.

During wafer sort, the test vectors are run on the chip using a probe card to supply signals and power to the device. This is far from an ideal electrical environment; if a large number of 1/0s were to switch simultaneously, it is possible that the resulting current surge could affect the internal state of the chip. To avoid this, additional inputs were included that would selectively disable a number of outputs. However, the vector set was large (150k vectors), and repeating the vector set with various combinations of outputs enabled would have made multiple copies of the vector set, which would have been unmanageable. An alternative approach was used in conjunction with Vitesse whereby one vector set would be used but the 'outputs would be selectively enabled on the tester.

Conclusion

The design of the Block V Receiver ASICs could certainly be called a challenge,

but the task was ultimately not a great deal more complex than designing for other technologies. While the GaAs gate array posed some problems with placement and testability, the design was completed with standard workstation tools, supplemented by other CAE products and in-house tools. Eight designers completed the project in about one year. The use of high-density gate arrays was essential for this application. There were no other methods with which to obtain the necessary density and speed while adhering to a tight design schedule. At the present time, the first chip (Jetcar) has been delivered, and is performing well.

ACKNOWLEDGMENT

The work described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Table 1. Chip Set Design Tools

Tool	Supplier	Function
GED	Cadence (Valid)	Design Entry
Rapidsim	Cadence (Valid)	Logic Simulation
VSCVR	Vitesse	Valid to VSC translator
VSCERC	Vitesse	ERC checker
VALID_BA	Vitesse	Valid back annotation
VSCTVED	Vitesse	Test vector checker
VR2MTV	Vitesse	Vitesse to Motive translator
VR2LASAR	Vitesse	Vitesse to Lasar translator
VSCGPRE	Vitesse	Graphical pre-placement
LASAR	Teradyne	Min/Max logic simulator
XP-140	Zycad	Simulation accelerator
SPW	Comdisco	Signal Processing Workstation
MOTIVE	Quad Design	Static Timing Analyzer
TESTGEN	Sunrise	Testability analysis and ATPG
IFA	JPL	Intelligent Front Annotation
EXAM	JPL	Statistical placement verifier
FLP	JPL	Fault list translator

photo

Figure 1. Deep Space Network 70-meter Antenna at Goldstone, California

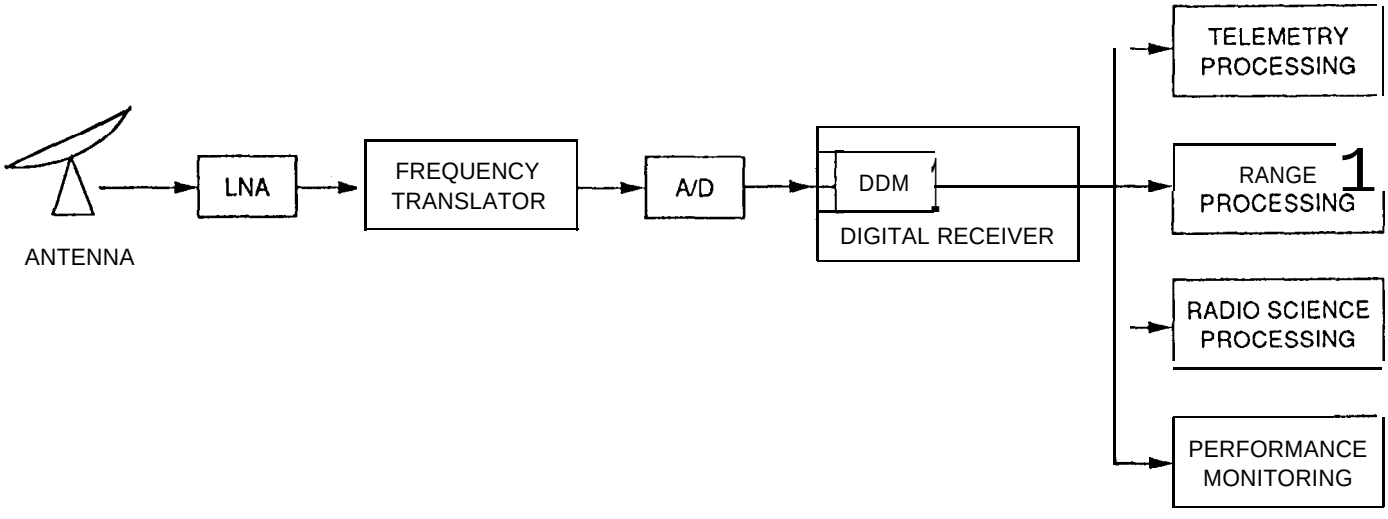

Figure 2. Block V Receiver

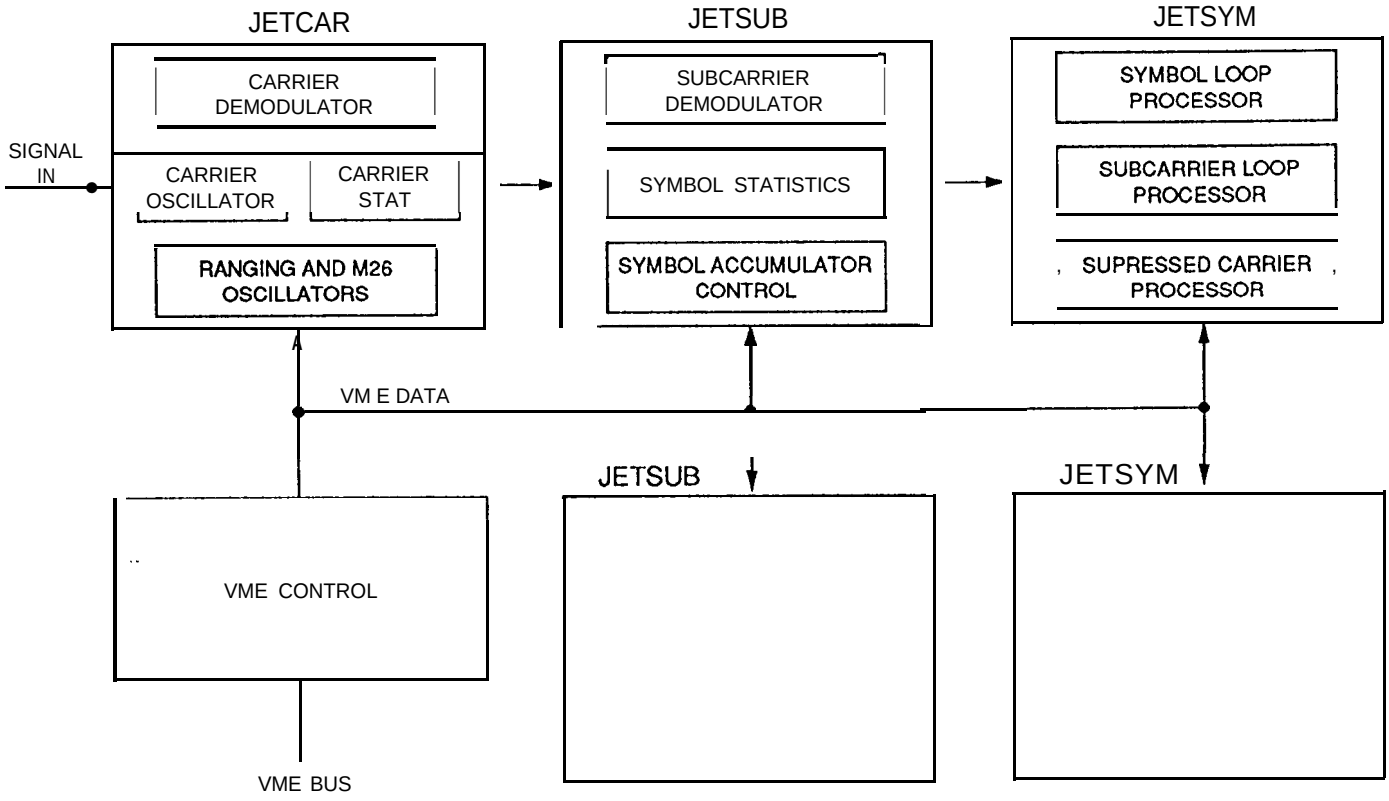

Figure 3. Digital Demodulator

(photo)

Figure.4 Jetcar Chip

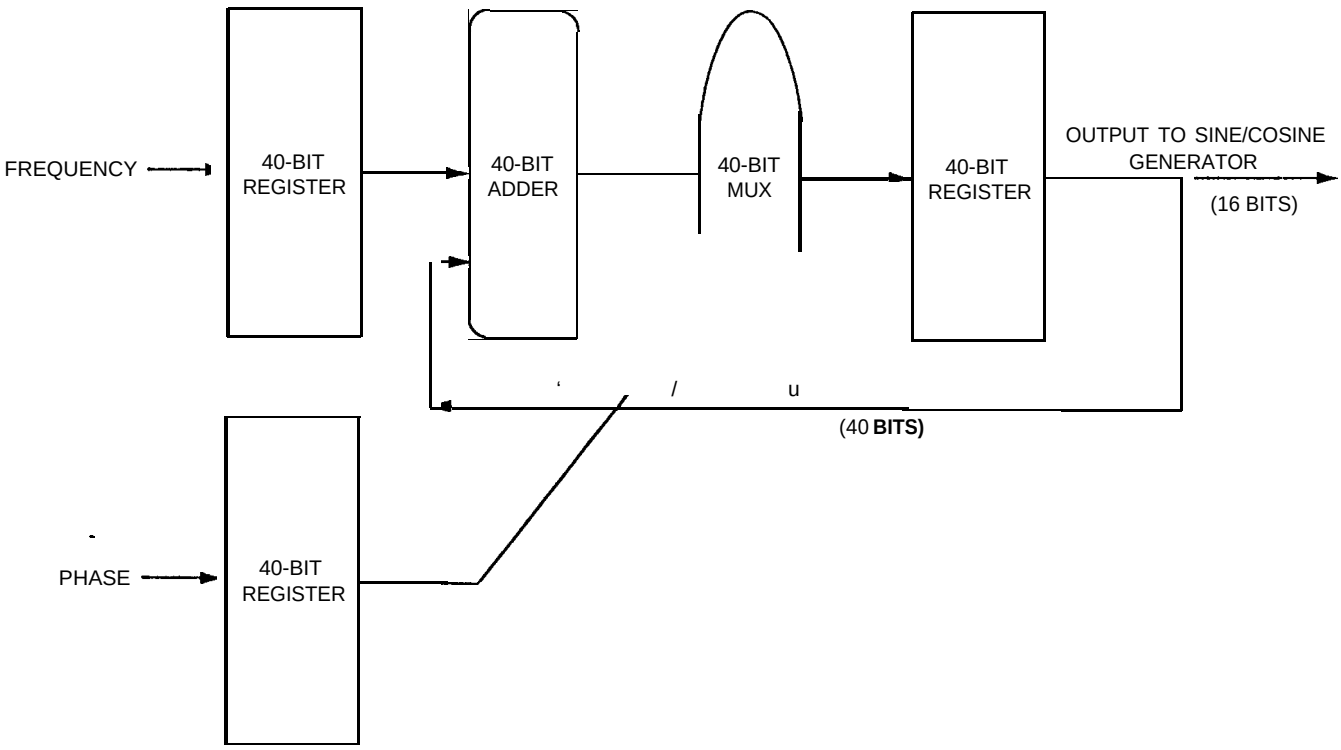

Figure 5. NCO Circuit

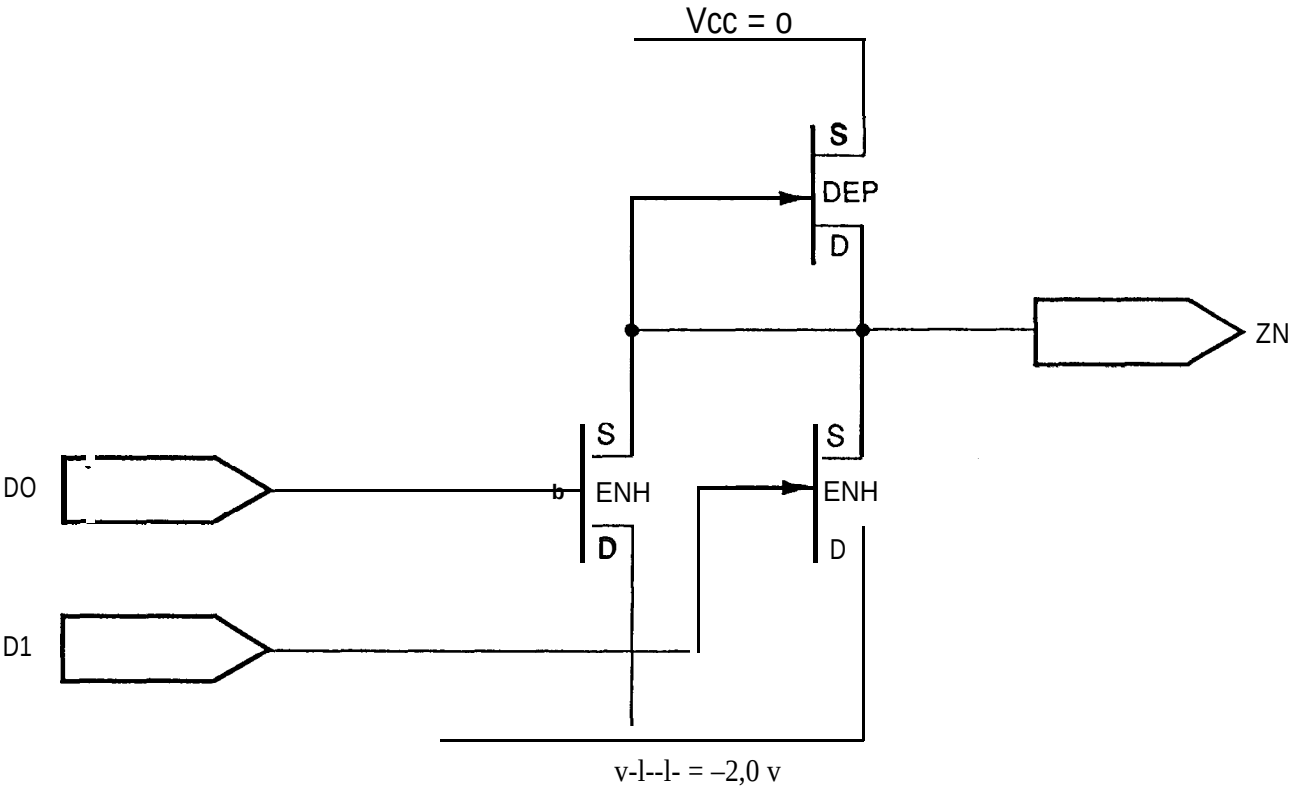

Figure 6. BASIC NOR GATE

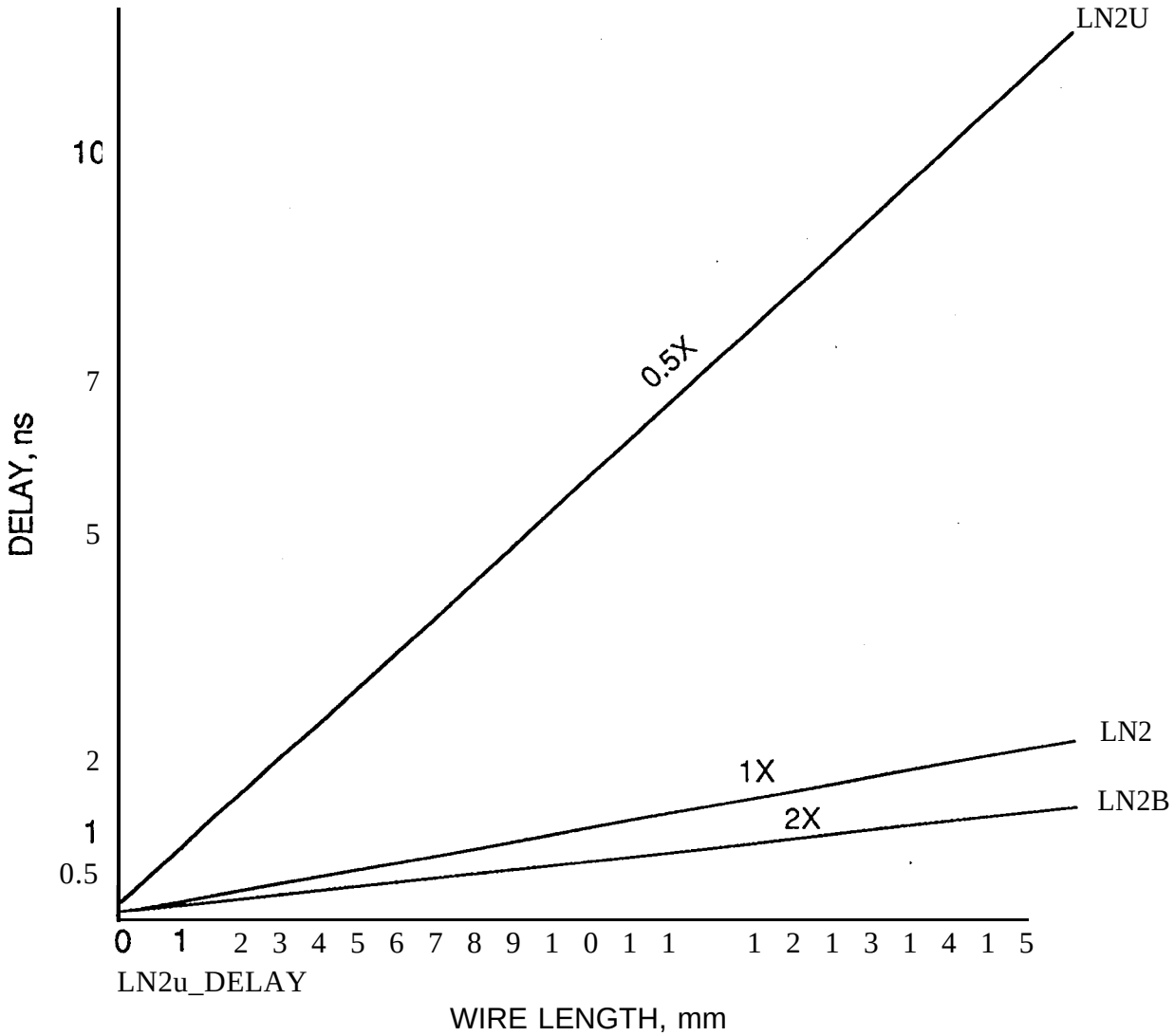

Figure 7. Effect of Wire Length on 2-i/p NOR Gate

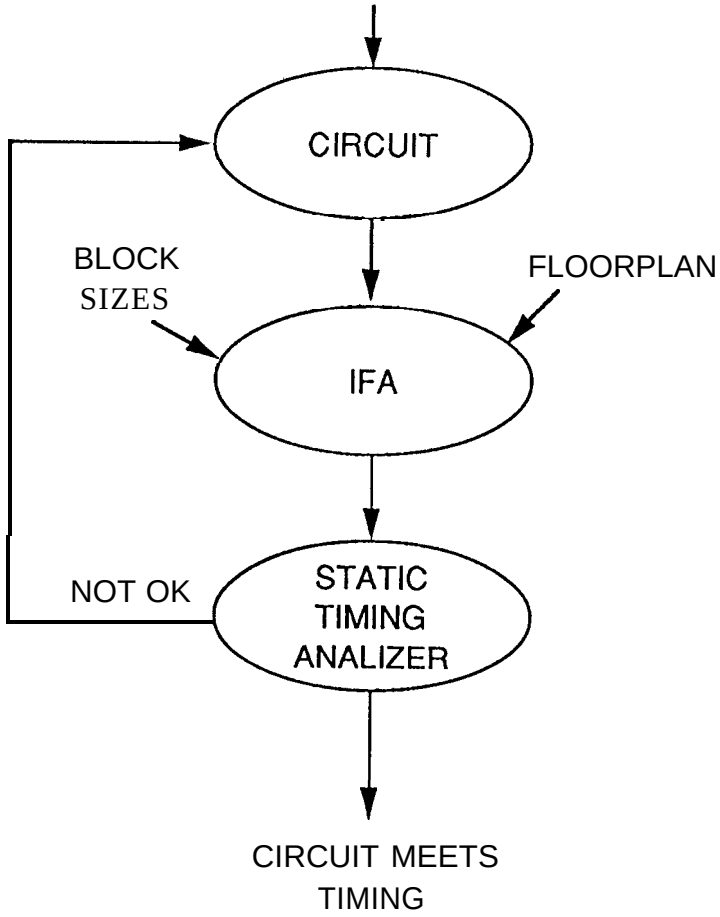

Figure 8. Verifying Timing

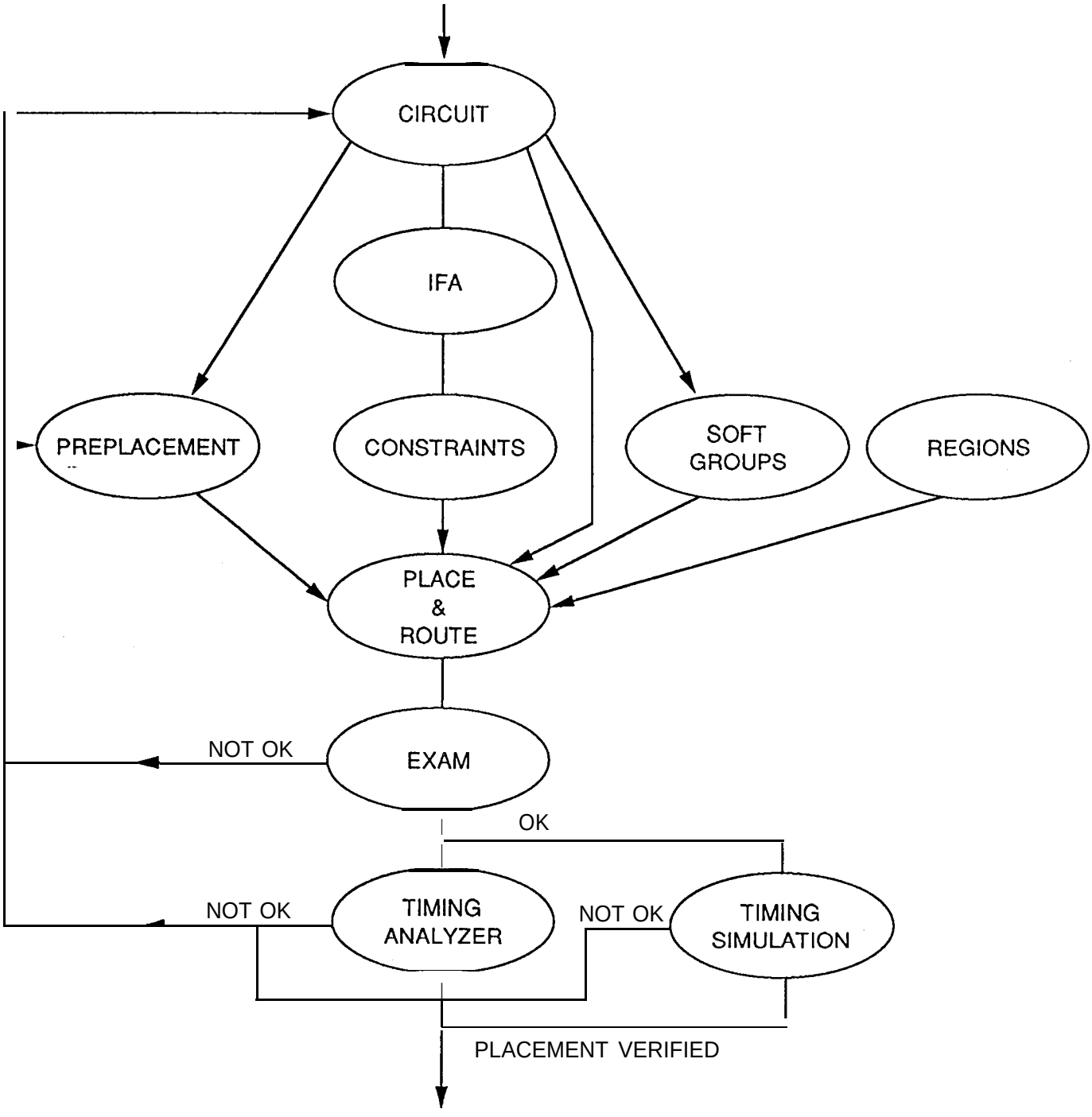

Figure 9. Placement

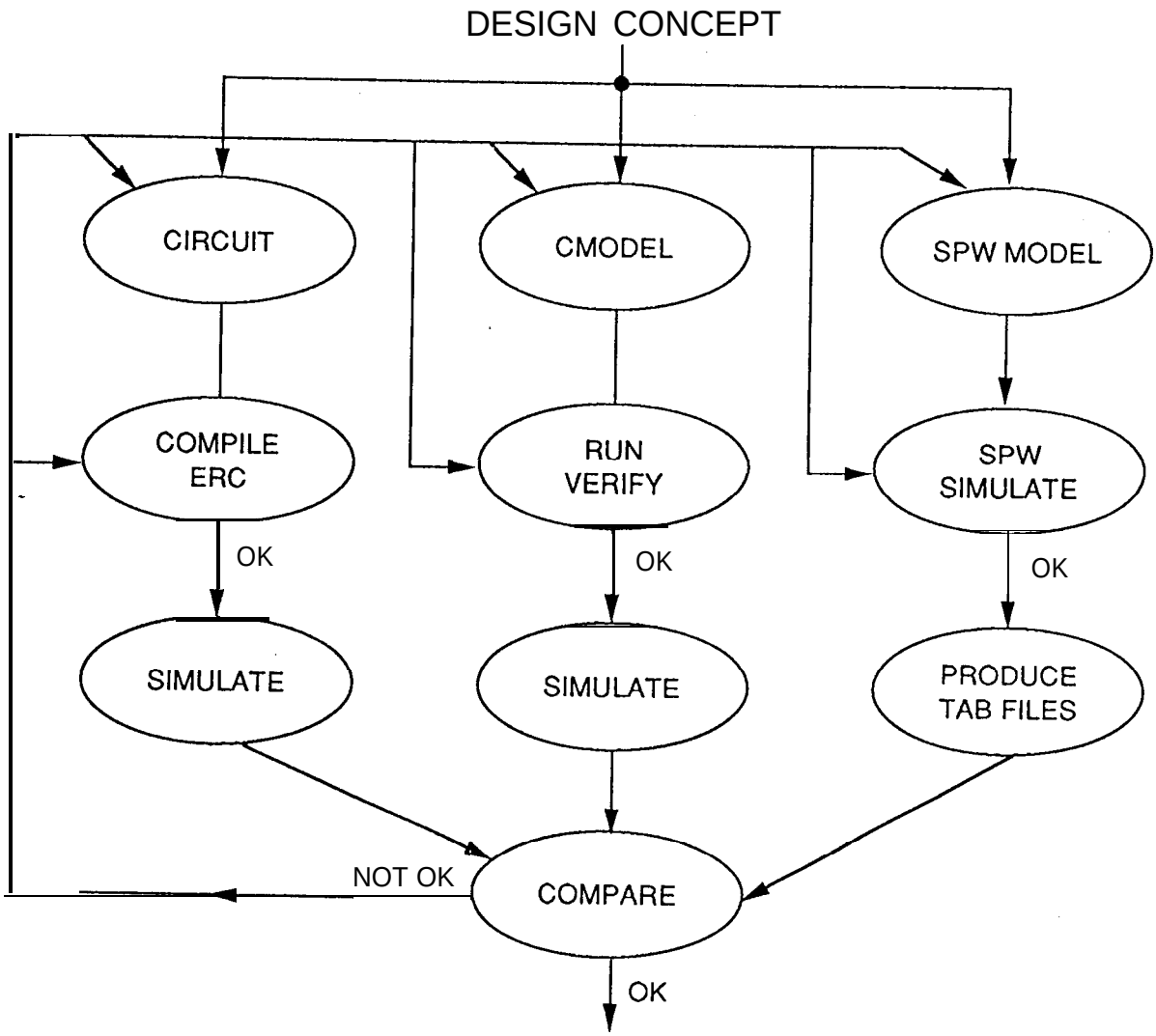

Figure 10. Design Process

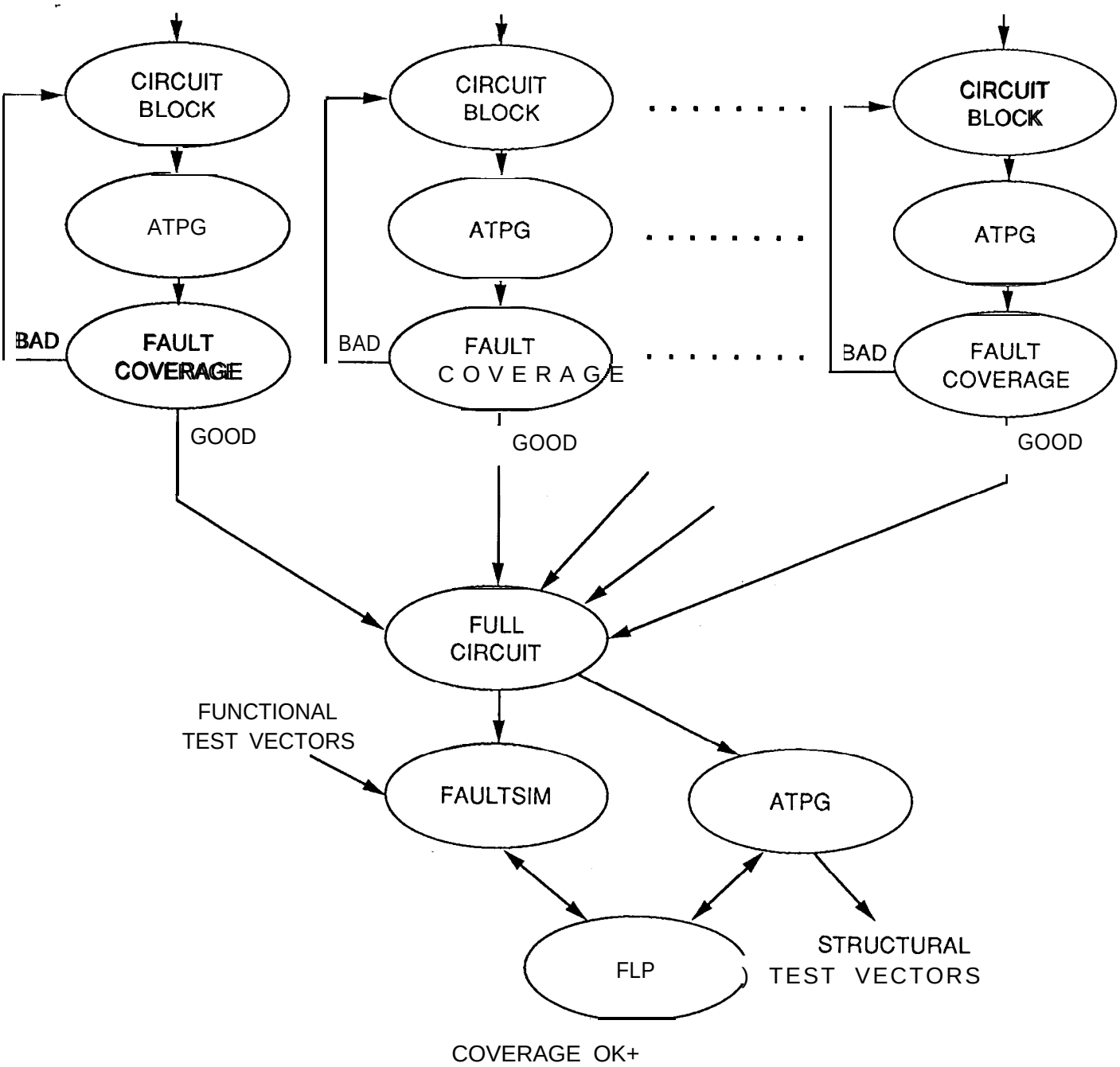

Figure 11. Test Coverage